**AUTHENTICATION VULNERABILITIES**

**BÀI 1: XÁC THỰC LÀ GÌ?**

Về mặt khái niệm, lỗ hổng xác thực dễ hiểu. Tuy nhiên, chúng thường rất quan trọng vì có mối quan hệ rõ ràng giữa xác thực và bảo mật.

Lỗ hổng xác thực có thể cho phép kẻ tấn công truy cập vào dữ liệu và chức năng nhạy cảm. Chúng cũng để lộ thêm bề mặt tấn công để khai thác thêm. Vì lý do này, điều quan trọng là phải tìm hiểu cách xác định và khai thác lỗ hổng xác thực và cách vượt qua các biện pháp bảo vệ thông thường.

Trong phần này, chúng tôi giải thích:

* Các cơ chế xác thực phổ biến nhất được các trang web sử dụng.
* Những lỗ hổng tiềm ẩn trong các cơ chế này.
* Các lỗ hổng cố hữu trong các cơ chế xác thực khác nhau.
* Các lỗ hổng điển hình phát sinh do việc triển khai không đúng cách.
* Làm thế nào để bạn có thể tạo cơ chế xác thực của riêng mình mạnh mẽ nhất có thể.

\* XÁC THỰC LÀ GÌ?

Xác thực là quá trình xác minh danh tính của người dùng hoặc khách hàng. Các trang web có khả năng bị lộ cho bất kỳ ai kết nối với internet. Điều này làm cho các cơ chế xác thực mạnh mẽ trở thành một phần không thể thiếu của bảo mật web hiệu quả.

Có ba loại xác thực chính:

* Một cái gì đó bạn biết , chẳng hạn như mật khẩu hoặc câu trả lời cho câu hỏi bảo mật. Đôi khi chúng được gọi là "yếu tố kiến ​​thức".
* Một thứ gì đó bạn có , Đây là một vật thể vật lý như điện thoại di động hoặc mã thông báo bảo mật. Đôi khi chúng được gọi là "yếu tố sở hữu".
* Một cái gì đó bạn là hoặc làm. Ví dụ, sinh trắc học hoặc mô hình hành vi của bạn. Đôi khi chúng được gọi là "yếu tố vốn có".

Cơ chế xác thực dựa vào nhiều công nghệ khác nhau để xác minh một hoặc nhiều yếu tố này.

- SỰ KHÁC BIỆT GIỮA XÁC THỰC VÀ ỦY QUYỀN LÀ GÌ?

Xác thực là quá trình xác minh người dùng có phải là người mà họ tuyên bố hay không. Ủy quyền liên quan đến việc xác minh xem người dùng có được phép làm điều gì đó hay không.

Ví dụ, xác thực xác định xem người đang cố gắng truy cập vào một trang web bằng tên người dùng Carlos123có thực sự là người đã tạo tài khoản hay không.

Sau khi Carlos123được xác thực, quyền của họ sẽ xác định những gì họ được phép làm. Ví dụ, họ có thể được phép truy cập thông tin cá nhân về người dùng khác hoặc thực hiện các hành động như xóa tài khoản của người dùng khác.

\* LỖ HỔNG XÁC THỰC PHÁT SINH THẾ NÀO?

Hầu hết các lỗ hổng trong cơ chế xác thực xảy ra theo một trong hai cách sau:

* Cơ chế xác thực yếu vì chúng không bảo vệ hiệu quả trước các cuộc tấn công bằng phương pháp dò mật khẩu.
* Lỗi logic hoặc mã hóa kém trong quá trình triển khai cho phép kẻ tấn công bỏ qua hoàn toàn cơ chế xác thực. Đôi khi điều này được gọi là "xác thực bị hỏng".

Trong nhiều lĩnh vực phát triển web, lỗi logic khiến trang web hoạt động không như mong đợi, có thể là vấn đề bảo mật hoặc không. Tuy nhiên, vì xác thực rất quan trọng đối với bảo mật, nên rất có thể logic xác thực bị lỗi sẽ khiến trang web gặp phải các vấn đề bảo mật.

\* TÁC ĐỘNG CỦA VIỆC XÁC THỰC DỄ BỊ TẤN CÔNG LÀ GÌ?

Tác động của lỗ hổng xác thực có thể rất nghiêm trọng. Nếu kẻ tấn công bỏ qua xác thực hoặc dùng vũ lực để xâm nhập vào tài khoản của người dùng khác, chúng có thể truy cập vào tất cả dữ liệu và chức năng mà tài khoản bị xâm phạm có. Nếu chúng có thể xâm phạm một tài khoản có đặc quyền cao, chẳng hạn như quản trị viên hệ thống, chúng có thể kiểm soát hoàn toàn toàn bộ ứng dụng và có khả năng truy cập vào cơ sở hạ tầng nội bộ.

Ngay cả khi xâm phạm một tài khoản có đặc quyền thấp vẫn có thể cấp cho kẻ tấn công quyền truy cập vào dữ liệu mà chúng không nên có, chẳng hạn như thông tin kinh doanh nhạy cảm về mặt thương mại. Ngay cả khi tài khoản không có quyền truy cập vào bất kỳ dữ liệu nhạy cảm nào, nó vẫn có thể cho phép kẻ tấn công truy cập vào các trang bổ sung, cung cấp thêm bề mặt tấn công. Thông thường, các cuộc tấn công có mức độ nghiêm trọng cao không thể thực hiện được từ các trang có thể truy cập công khai, nhưng chúng có thể thực hiện được từ một trang nội bộ.

\* LỖI TRONG CƠ CHẾ XÁC THỰC

Hệ thống xác thực của trang web thường bao gồm một số cơ chế riêng biệt có thể xảy ra lỗ hổng. Một số lỗ hổng có thể áp dụng cho tất cả các bối cảnh này. Một số khác cụ thể hơn đối với chức năng được cung cấp.

Chúng tôi sẽ xem xét kỹ hơn một số lỗ hổng phổ biến nhất trong các lĩnh vực sau:

* Lỗ hổng trong LABS đăng nhập bằng mật khẩu
* Lỗ hổng trong xác thực đa yếu tố LABS
* Các lỗ hổng trong các cơ chế xác thực khác LABS

Một số phòng thí nghiệm yêu cầu bạn liệt kê tên người dùng và mật khẩu brute-force. Để giúp bạn thực hiện quy trình này, chúng tôi cung cấp danh sách ngắn các tên người dùng và mật khẩu ứng viên mà bạn nên sử dụng để giải các phòng thí nghiệm.

- LỖ HỔNG TRONG CƠ CHẾ XÁC THỰC CỦA BÊN THỨ 3:

Nếu bạn thích hack cơ chế xác thực và đã hoàn tất quá trình xác thực chính, bạn có thể muốn thử phòng thí nghiệm xác thực OAuth của chúng tôi.

**BÀI 2: LỖ HỔNG TRONG ĐĂNG NHẬP BẰNG MẬT KHẨU**

Đối với các trang web áp dụng quy trình đăng nhập bằng mật khẩu, người dùng tự đăng ký tài khoản hoặc được quản trị viên chỉ định một tài khoản. Tài khoản này được liên kết với tên người dùng duy nhất và mật khẩu bí mật, người dùng nhập vào biểu mẫu đăng nhập để xác thực.

Trong trường hợp này, việc họ biết mật khẩu bí mật được coi là bằng chứng đủ để xác định danh tính của người dùng. Điều này có nghĩa là tính bảo mật của trang web bị xâm phạm nếu kẻ tấn công có thể lấy hoặc đoán được thông tin đăng nhập của người dùng khác.

Có thể thực hiện điều này theo nhiều cách. Các phần sau đây sẽ chỉ ra cách kẻ tấn công có thể sử dụng các cuộc tấn công brute-force và một số lỗ hổng trong bảo vệ brute-force. Bạn cũng sẽ tìm hiểu về các lỗ hổng trong xác thực HTTP cơ bản.

\* TẤN CÔNG BẰNG BRUTE-FORCE

Tấn công brute-force là khi kẻ tấn công sử dụng hệ thống thử và sai để đoán thông tin đăng nhập hợp lệ của người dùng. Các cuộc tấn công này thường được tự động hóa bằng cách sử dụng danh sách từ của tên người dùng và mật khẩu. Tự động hóa quy trình này, đặc biệt là sử dụng các công cụ chuyên dụng, có khả năng cho phép kẻ tấn công thực hiện số lượng lớn các lần đăng nhập với tốc độ cao.

Brute-force không phải lúc nào cũng chỉ là trường hợp đoán tên người dùng và mật khẩu hoàn toàn ngẫu nhiên. Bằng cách sử dụng logic cơ bản hoặc kiến ​​thức công khai, kẻ tấn công có thể tinh chỉnh các cuộc tấn công brute-force để đưa ra những phỏng đoán có căn cứ hơn nhiều. Điều này làm tăng đáng kể hiệu quả của các cuộc tấn công như vậy. Các trang web dựa vào đăng nhập bằng mật khẩu làm phương pháp duy nhất để xác thực người dùng có thể rất dễ bị tấn công nếu không triển khai đủ biện pháp bảo vệ brute-force.

- TẤN CÔNG USERNAME KIỂU BRUTE-FORCE

Tên người dùng đặc biệt dễ đoán nếu chúng tuân theo một mẫu dễ nhận biết, chẳng hạn như địa chỉ email. Ví dụ, rất phổ biến khi thấy thông tin đăng nhập doanh nghiệp theo định dạng firstname.lastname@somecompany.com. Tuy nhiên, ngay cả khi không có mẫu rõ ràng, đôi khi ngay cả các tài khoản có đặc quyền cao cũng được tạo bằng tên người dùng có thể dự đoán được, chẳng hạn như adminhoặc administrator.

Trong quá trình kiểm tra, hãy kiểm tra xem trang web có tiết lộ tên người dùng tiềm năng công khai hay không. Ví dụ, bạn có thể truy cập hồ sơ người dùng mà không cần đăng nhập không? Ngay cả khi nội dung thực tế của hồ sơ bị ẩn, tên được sử dụng trong hồ sơ đôi khi giống với tên người dùng đăng nhập. Bạn cũng nên kiểm tra phản hồi HTTP để xem có bất kỳ địa chỉ email nào bị tiết lộ không. Thỉnh thoảng, phản hồi chứa địa chỉ email của người dùng có đặc quyền cao, chẳng hạn như quản trị viên hoặc hỗ trợ CNTT.

- TẤN CÔNG MẬT KHẨU KIỂU BRUTE-FORCE

Mật khẩu cũng có thể bị tấn công bằng cách dùng vũ lực, với độ khó thay đổi tùy theo độ mạnh của mật khẩu. Nhiều trang web áp dụng một số hình thức chính sách mật khẩu, buộc người dùng phải tạo mật khẩu có độ phức tạp cao, ít nhất là về mặt lý thuyết, khó bẻ khóa hơn chỉ bằng cách dùng vũ lực. Điều này thường liên quan đến việc thực thi mật khẩu bằng:

* Số lượng ký tự tối thiểu
* Một sự kết hợp của chữ thường và chữ hoa
* Ít nhất một ký tự đặc biệt

Tuy nhiên, trong khi mật khẩu entropy cao rất khó để máy tính có thể bẻ khóa, chúng ta có thể sử dụng kiến ​​thức cơ bản về hành vi của con người để khai thác các lỗ hổng mà người dùng vô tình đưa vào hệ thống này. Thay vì tạo mật khẩu mạnh với sự kết hợp ngẫu nhiên các ký tự, người dùng thường lấy một mật khẩu mà họ có thể nhớ và cố gắng bẻ khóa nó để phù hợp với chính sách mật khẩu. Ví dụ, nếu mypasswordkhông được phép, người dùng có thể thử một cái gì đó như Mypassword1! hoặc Myp4$$w0rd thay vào đó.

Trong trường hợp chính sách yêu cầu người dùng thay đổi mật khẩu thường xuyên, người dùng cũng thường chỉ thực hiện những thay đổi nhỏ, có thể dự đoán được đối với mật khẩu ưa thích của họ. Ví dụ, Mypassword1!trở thành Mypassword1? hoặc Mypassword2!.

Kiến thức về thông tin xác thực có thể có và các mẫu có thể dự đoán được này có nghĩa là các cuộc tấn công bằng cách dùng vũ lực thường tinh vi hơn nhiều và do đó hiệu quả hơn so với việc chỉ lặp lại mọi tổ hợp ký tự có thể có.

- LIỆT KÊ TÊN NGƯỜI DÙNG:

Liệt kê tên người dùng là khi kẻ tấn công có thể quan sát những thay đổi trong hành vi của trang web để xác định xem tên người dùng có hợp lệ hay không.

Việc liệt kê tên người dùng thường xảy ra trên trang đăng nhập, ví dụ, khi bạn nhập tên người dùng hợp lệ nhưng mật khẩu không đúng hoặc trên biểu mẫu đăng ký khi bạn nhập tên người dùng đã được sử dụng. Điều này làm giảm đáng kể thời gian và công sức cần thiết để tấn công bằng cách dùng vũ lực để đăng nhập vì kẻ tấn công có thể nhanh chóng tạo danh sách rút gọn các tên người dùng hợp lệ.

Khi cố gắng tấn công bằng cách dùng vũ lực để mở trang đăng nhập, bạn nên đặc biệt chú ý đến bất kỳ điểm khác biệt nào trong:

* Mã trạng thái : Trong một cuộc tấn công brute-force, mã trạng thái HTTP trả về có khả năng giống nhau đối với phần lớn các phỏng đoán vì hầu hết chúng sẽ sai. Nếu một phỏng đoán trả về mã trạng thái khác, thì đây là dấu hiệu rõ ràng cho thấy tên người dùng là đúng. Thực hành tốt nhất đối với các trang web là luôn trả về cùng một mã trạng thái bất kể kết quả ra sao, nhưng thực hành này không phải lúc nào cũng được tuân theo.
* Thông báo lỗi : Đôi khi thông báo lỗi trả về khác nhau tùy thuộc vào việc cả tên người dùng VÀ mật khẩu đều không đúng hay chỉ có mật khẩu không đúng. Thực hành tốt nhất cho các trang web là sử dụng các thông báo chung, giống hệt nhau trong cả hai trường hợp, nhưng đôi khi vẫn có lỗi đánh máy nhỏ. Chỉ cần một ký tự không đúng chỗ cũng khiến hai thông báo khác biệt, ngay cả trong trường hợp ký tự không hiển thị trên trang được hiển thị.
* Thời gian phản hồi : Nếu hầu hết các yêu cầu được xử lý với thời gian phản hồi tương tự, bất kỳ yêu cầu nào khác với thời gian này đều cho thấy có điều gì đó khác đang diễn ra ở hậu trường. Đây là một dấu hiệu khác cho thấy tên người dùng được đoán có thể đúng. Ví dụ: một trang web chỉ có thể kiểm tra xem mật khẩu có đúng không nếu tên người dùng hợp lệ. Bước bổ sung này có thể khiến thời gian phản hồi tăng nhẹ. Điều này có thể tinh vi, nhưng kẻ tấn công có thể khiến sự chậm trễ này trở nên rõ ràng hơn bằng cách nhập mật khẩu quá dài khiến trang web mất nhiều thời gian hơn để xử lý.

\* BẢO VỆ BRUCE-FORCE BROKEN

Rất có khả năng một cuộc tấn công brute-force sẽ liên quan đến nhiều lần đoán sai trước khi kẻ tấn công xâm phạm thành công một tài khoản. Về mặt logic, bảo vệ brute-force xoay quanh việc cố gắng làm cho quá trình tự động hóa và làm chậm tốc độ mà kẻ tấn công có thể cố gắng đăng nhập trở nên khó khăn nhất có thể. Hai cách phổ biến nhất để ngăn chặn các cuộc tấn công brute-force là:

* Khóa tài khoản mà người dùng từ xa đang cố gắng truy cập nếu họ thực hiện quá nhiều lần đăng nhập không thành công
* Chặn địa chỉ IP của người dùng từ xa nếu họ thực hiện quá nhiều lần đăng nhập liên tiếp

Cả hai cách tiếp cận đều cung cấp các mức độ bảo vệ khác nhau, nhưng không cách nào là bất khả xâm phạm, đặc biệt là nếu được triển khai bằng logic sai lầm.

Ví dụ, đôi khi bạn có thể thấy rằng IP của mình bị chặn nếu bạn không đăng nhập quá nhiều lần. Trong một số triển khai, bộ đếm số lần thử không thành công sẽ được đặt lại nếu chủ sở hữu IP đăng nhập thành công. Điều này có nghĩa là kẻ tấn công chỉ cần đăng nhập vào tài khoản của chính họ sau mỗi vài lần thử để ngăn chặn giới hạn này không bao giờ đạt đến.

Trong trường hợp này, chỉ cần thêm thông tin đăng nhập của bạn vào danh sách từ theo các khoảng thời gian đều đặn là đủ để khiến biện pháp phòng thủ này trở nên vô dụng.

- KHÓA TÀI KHOẢN:

Một cách mà các trang web cố gắng ngăn chặn brute-forcing là khóa tài khoản nếu đáp ứng một số tiêu chí đáng ngờ, thường là một số lần đăng nhập không thành công. Cũng giống như lỗi đăng nhập thông thường, phản hồi từ máy chủ cho biết tài khoản đã bị khóa cũng có thể giúp kẻ tấn công liệt kê tên người dùng.

Khóa tài khoản cung cấp một mức độ bảo vệ nhất định chống lại việc tấn công brute-force nhắm mục tiêu vào một tài khoản cụ thể. Tuy nhiên, cách tiếp cận này không ngăn chặn đầy đủ các cuộc tấn công brute-force trong đó kẻ tấn công chỉ cố gắng truy cập vào bất kỳ tài khoản ngẫu nhiên nào mà chúng có thể.

Ví dụ, phương pháp sau đây có thể được sử dụng để giải quyết loại bảo vệ này:

1. Thiết lập danh sách tên người dùng ứng viên có khả năng hợp lệ. Có thể thông qua việc liệt kê tên người dùng hoặc chỉ dựa trên danh sách tên người dùng phổ biến.
2. Quyết định một danh sách rất ngắn các mật khẩu mà bạn nghĩ rằng ít nhất một người dùng có thể có. Quan trọng là, số lượng mật khẩu bạn chọn không được vượt quá số lần đăng nhập được phép. Ví dụ, nếu bạn đã tính toán được giới hạn là 3 lần thử, bạn cần chọn tối đa 3 lần đoán mật khẩu.
3. Sử dụng một công cụ như Burp Intruder, hãy thử từng mật khẩu đã chọn với từng tên người dùng ứng viên. Theo cách này, bạn có thể thử brute-force mọi tài khoản mà không kích hoạt khóa tài khoản. Bạn chỉ cần một người dùng duy nhất sử dụng một trong ba mật khẩu để xâm phạm một tài khoản.

Khóa tài khoản cũng không bảo vệ được chống lại các cuộc tấn công nhồi thông tin xác thực. Điều này liên quan đến việc sử dụng một từ điển lớn username:password gồm các cặp thông tin xác thực đăng nhập bị đánh cắp trong các vụ vi phạm dữ liệu. Nhồi thông tin xác thực dựa trên thực tế là nhiều người sử dụng lại cùng một tên người dùng và mật khẩu trên nhiều trang web và do đó, có khả năng một số thông tin xác thực bị xâm phạm trong từ điển cũng hợp lệ trên trang web mục tiêu. Khóa tài khoản không bảo vệ được chống lại nhồi thông tin xác thực vì mỗi tên người dùng chỉ được thử một lần. Nhồi thông tin xác thực đặc biệt nguy hiểm vì đôi khi nó có thể khiến kẻ tấn công xâm phạm nhiều tài khoản khác nhau chỉ bằng một cuộc tấn công tự động duy nhất.

- GIỚI HẠN TỶ LỆ NGƯỜI DÙNG:

Một cách khác mà các trang web cố gắng ngăn chặn các cuộc tấn công brute-force là thông qua giới hạn tỷ lệ người dùng. Trong trường hợp này, việc thực hiện quá nhiều yêu cầu đăng nhập trong một khoảng thời gian ngắn sẽ khiến địa chỉ IP của bạn bị chặn. Thông thường, IP chỉ có thể được bỏ chặn theo một trong các cách sau:

* Tự động sau khi một khoảng thời gian nhất định đã trôi qua
* Được thực hiện thủ công bởi người quản trị
* Thực hiện thủ công bởi người dùng sau khi hoàn thành thành công CAPTCHA

Giới hạn tỷ lệ người dùng đôi khi được ưa chuộng hơn khóa tài khoản vì ít bị liệt kê tên người dùng và tấn công từ chối dịch vụ. Tuy nhiên, nó vẫn chưa hoàn toàn an toàn. Như chúng ta đã thấy trong một ví dụ trong phòng thí nghiệm trước đó, có một số cách kẻ tấn công có thể thao túng IP rõ ràng của chúng để vượt qua lệnh chặn.

Vì giới hạn dựa trên tốc độ yêu cầu HTTP được gửi từ địa chỉ IP của người dùng nên đôi khi cũng có thể bỏ qua biện pháp phòng thủ này nếu bạn có thể tìm ra cách đoán nhiều mật khẩu chỉ bằng một yêu cầu.

\* XÁC THỰC CƠ BẢN HTTP:

Mặc dù khá cũ, nhưng tính đơn giản và dễ triển khai tương đối của nó có nghĩa là đôi khi bạn có thể thấy xác thực cơ bản HTTP được sử dụng. Trong xác thực cơ bản HTTP, máy khách nhận được mã thông báo xác thực từ máy chủ, được xây dựng bằng cách nối tên người dùng và mật khẩu, và mã hóa nó trong Base64. Mã thông báo này được lưu trữ và quản lý bởi trình duyệt, trình duyệt tự động thêm nó vào Authorizationtiêu đề của mọi yêu cầu tiếp theo như sau:

Authorization: Basic base64(username:password)

Vì một số lý do, phương pháp này thường không được coi là phương pháp xác thực an toàn. Đầu tiên, nó liên quan đến việc gửi đi nhiều lần thông tin đăng nhập của người dùng với mọi yêu cầu. Trừ khi trang web cũng triển khai HSTS, thông tin đăng nhập của người dùng có thể bị đánh cắp trong một cuộc tấn công trung gian.

Ngoài ra, việc triển khai xác thực HTTP cơ bản thường không hỗ trợ bảo vệ bằng cách tấn công kiểu brute-force. Vì mã thông báo chỉ bao gồm các giá trị tĩnh nên điều này có thể khiến nó dễ bị tấn công bằng cách brute-force.

Xác thực cơ bản HTTP cũng đặc biệt dễ bị khai thác liên quan đến phiên, đặc biệt là CSRF, mà bản thân nó không có khả năng bảo vệ.

Trong một số trường hợp, việc khai thác xác thực cơ bản HTTP dễ bị tấn công chỉ có thể cấp cho kẻ tấn công quyền truy cập vào một trang có vẻ không thú vị. Tuy nhiên, ngoài việc cung cấp thêm bề mặt tấn công, thông tin xác thực bị lộ theo cách này có thể được sử dụng lại trong các bối cảnh khác, bảo mật hơn.

**BÀI 3: LỖ HỔNG TRONG XÁC THỨC ĐA YẾU TỐ**

Trong phần này, chúng ta sẽ xem xét một số lỗ hổng có thể xảy ra trong cơ chế xác thực đa yếu tố. Chúng tôi cũng cung cấp một số phòng thí nghiệm tương tác để chứng minh cách bạn có thể khai thác các lỗ hổng này trong xác thực đa yếu tố.

Nhiều trang web chỉ dựa vào xác thực một yếu tố bằng mật khẩu để xác thực người dùng. Tuy nhiên, một số trang web yêu cầu người dùng chứng minh danh tính của mình bằng nhiều yếu tố xác thực.

Xác minh các yếu tố sinh trắc học là không thực tế đối với hầu hết các trang web. Tuy nhiên, ngày càng phổ biến khi thấy cả xác thực hai yếu tố bắt buộc và tùy chọn (2FA) dựa trên thứ bạn biết và thứ bạn có . Điều này thường yêu cầu người dùng nhập cả mật khẩu truyền thống và mã xác minh tạm thời từ thiết bị vật lý ngoài băng tần mà họ sở hữu.

Mặc dù đôi khi kẻ tấn công có thể lấy được một yếu tố dựa trên kiến ​​thức duy nhất, chẳng hạn như mật khẩu, nhưng khả năng có thể đồng thời lấy được một yếu tố khác từ một nguồn ngoài băng tần là rất thấp. Vì lý do này, xác thực hai yếu tố rõ ràng an toàn hơn xác thực một yếu tố. Tuy nhiên, giống như bất kỳ biện pháp bảo mật nào, nó chỉ an toàn như cách triển khai của nó. Xác thực hai yếu tố được triển khai kém có thể bị đánh bại hoặc thậm chí bị bỏ qua hoàn toàn, giống như xác thực một yếu tố.

Cũng đáng lưu ý rằng lợi ích đầy đủ của xác thực đa yếu tố chỉ đạt được bằng cách xác minh nhiều yếu tố khác nhau . Xác minh cùng một yếu tố theo hai cách khác nhau không phải là xác thực hai yếu tố thực sự. 2FA dựa trên email là một ví dụ như vậy. Mặc dù người dùng phải cung cấp mật khẩu và mã xác minh, nhưng việc truy cập mã chỉ dựa vào thông tin đăng nhập cho tài khoản email của họ. Do đó, yếu tố xác thực kiến ​​thức chỉ đơn giản là được xác minh hai lần.

\* MÃ THÔNG BÁO XÁC THỰC 2 YẾU TỐ

Mã xác minh thường được người dùng đọc từ một thiết bị vật lý nào đó. Nhiều trang web bảo mật cao hiện cung cấp cho người dùng một thiết bị chuyên dụng cho mục đích này, chẳng hạn như mã thông báo RSA hoặc thiết bị bàn phím mà bạn có thể sử dụng để truy cập ngân hàng trực tuyến hoặc máy tính xách tay làm việc của mình. Ngoài việc được thiết kế riêng cho mục đích bảo mật, các thiết bị chuyên dụng này còn có lợi thế là tạo mã xác minh trực tiếp. Các trang web cũng thường sử dụng ứng dụng di động chuyên dụng, chẳng hạn như Google Authenticator, vì lý do tương tự.

Mặt khác, một số trang web gửi mã xác minh đến điện thoại di động của người dùng dưới dạng tin nhắn văn bản. Mặc dù về mặt kỹ thuật, điều này vẫn xác minh yếu tố "thứ bạn có", nhưng nó có thể bị lạm dụng. Đầu tiên, mã được truyền qua SMS thay vì được tạo bởi chính thiết bị. Điều này tạo ra khả năng mã bị chặn. Ngoài ra còn có nguy cơ hoán đổi SIM, trong đó kẻ tấn công gian lận lấy được thẻ SIM có số điện thoại của nạn nhân. Sau đó, kẻ tấn công sẽ nhận được tất cả tin nhắn SMS được gửi đến nạn nhân, bao gồm cả tin nhắn có chứa mã xác minh của họ.

\* BỎ QUA XÁC THỰC 2 YẾU TỐ

Đôi khi, việc triển khai xác thực hai yếu tố bị lỗi đến mức có thể bị bỏ qua hoàn toàn.

Nếu người dùng được nhắc nhập mật khẩu trước, sau đó được nhắc nhập mã xác minh trên một trang riêng, thì người dùng thực sự ở trạng thái "đã đăng nhập" trước khi họ nhập mã xác minh. Trong trường hợp này, bạn nên thử xem bạn có thể trực tiếp bỏ qua các trang "chỉ đăng nhập" sau khi hoàn tất bước xác thực đầu tiên hay không. Thỉnh thoảng, bạn sẽ thấy rằng một trang web không thực sự kiểm tra xem bạn đã hoàn tất bước thứ hai hay chưa trước khi tải trang.

\* LOGIC XÁC MINH 2 YẾU TỐ BỊ LỖI

Đôi khi, logic sai lầm trong xác thực hai yếu tố có nghĩa là sau khi người dùng hoàn tất bước đăng nhập đầu tiên, trang web không xác minh đầy đủ rằng người dùng đó đang hoàn tất bước thứ hai.

Ví dụ, người dùng đăng nhập bằng thông tin đăng nhập thông thường của họ ở bước đầu tiên như sau:

POST /login-steps/first HTTP/1.1

Host: vulnerable-website.com

...

username=carlos&password=qwerty

Sau đó, họ được chỉ định một cookie liên quan đến tài khoản của họ trước khi chuyển sang bước thứ hai của quy trình đăng nhập:

HTTP/1.1 200 OK

Set-Cookie: account=carlos

GET /login-steps/second HTTP/1.1

Cookie: account=carlos

Khi gửi mã xác minh, yêu cầu sẽ sử dụng cookie này để xác định tài khoản nào người dùng đang cố truy cập:

POST /login-steps/second HTTP/1.1

Host: vulnerable-website.com

Cookie: account=carlos

...

verification-code=123456

Trong trường hợp này, kẻ tấn công có thể đăng nhập bằng thông tin đăng nhập của riêng chúng nhưng sau đó thay đổi giá trị của accountcookie thành bất kỳ tên người dùng tùy ý nào khi gửi mã xác minh.

POST /login-steps/second HTTP/1.1

Host: vulnerable-website.com

Cookie: account=victim-user

...

verification-code=123456

Điều này cực kỳ nguy hiểm nếu kẻ tấn công sau đó có thể dùng brute-force để lấy mã xác minh vì nó sẽ cho phép chúng đăng nhập vào tài khoản của người dùng tùy ý hoàn toàn dựa trên tên người dùng của họ. Chúng thậm chí không cần biết mật khẩu của người dùng.

**BÀI 4: LỖ HỔNG TRONG CƠ CHẾ XÁC THỰC KHÁC**

Ngoài ra, cơ sở đăng nhập chức năng, hầu hết các trang web đều cung cấp chức năng bổ sung để cho phép người dùng quản lý tài khoản của họ. Ví dụ: người dùng thường có thể thay đổi mật khẩu hoặc đặt lại mật khẩu khi quên. Cơ chế này cũng có thể tạo ra các ổ kho có thể tấn công khai thác thác thể.

Các trang web thường cẩn thận để tránh các ổ phổ biến trong trang đăng nhập của họ. Nhưng thật dễ dàng để bỏ qua thực tế là bạn cần thực hiện các bước tương tự để đảm bảo rằng chức năng liên quan cũng mạnh mẽ như nhau. Điều đặc biệt quan trọng trong trường hợp tấn công có thể tạo tài khoản của riêng mình và do đó có thể dễ dàng truy cập để nghiên cứu các trang bổ sung này.

\* GIỮ NGƯỜI DÙNG ĐĂNG NHẬP

Một tính năng phổ biến là tùy chọn duy trì trạng thái đăng nhập ngay cả sau khi đóng phiên trình duyệt. Đây thường là một hộp kiểm đơn giản có nhãn như "Ghi nhớ tôi" hoặc "Giữ tôi đăng nhập".

Chức năng này thường được triển khai bằng cách tạo mã thông báo "ghi nhớ tôi" nào đó, sau đó được lưu trữ trong cookie cố định. Vì sở hữu cookie này cho phép bạn bỏ qua toàn bộ quy trình đăng nhập nên cách tốt nhất là không nên đoán cookie này. Tuy nhiên, một số trang web tạo cookie này dựa trên chuỗi giá trị tĩnh có thể dự đoán được, chẳng hạn như tên người dùng và dấu thời gian. Một số thậm chí còn sử dụng mật khẩu làm một phần của cookie. Cách tiếp cận này đặc biệt nguy hiểm nếu kẻ tấn công có thể tạo tài khoản của riêng họ vì họ có thể nghiên cứu cookie của riêng mình và có khả năng suy ra cách tạo ra cookie đó. Sau khi tìm ra công thức, họ có thể thử tấn công bằng vũ lực cookie của người dùng khác để truy cập vào tài khoản của họ.

Một số trang web cho rằng nếu cookie được mã hóa theo một cách nào đó thì sẽ không thể đoán được ngay cả khi cookie đó sử dụng các giá trị tĩnh. Mặc dù điều này có thể đúng nếu thực hiện đúng, nhưng việc "mã hóa" cookie một cách ngây thơ bằng cách sử dụng mã hóa hai chiều đơn giản như Base64 không mang lại bất kỳ sự bảo vệ nào. Ngay cả khi sử dụng mã hóa phù hợp với hàm băm một chiều cũng không hoàn toàn an toàn. Nếu kẻ tấn công có thể dễ dàng xác định thuật toán băm và không sử dụng muối, chúng có khả năng tấn công bằng vũ lực cookie chỉ bằng cách băm danh sách từ của chúng. Phương pháp này có thể được sử dụng để bỏ qua giới hạn số lần thử đăng nhập nếu không áp dụng giới hạn tương tự cho các lần đoán cookie.

Ngay cả khi kẻ tấn công không thể tạo tài khoản của riêng mình, chúng vẫn có thể khai thác lỗ hổng này. Sử dụng các kỹ thuật thông thường, chẳng hạn như XSS, kẻ tấn công có thể đánh cắp cookie "ghi nhớ tôi" của người dùng khác và suy ra cách cookie được xây dựng từ đó. Nếu trang web được xây dựng bằng khuôn khổ mã nguồn mở, các chi tiết chính về cấu trúc cookie thậm chí có thể được ghi lại công khai.

Trong một số trường hợp hiếm hoi, có thể lấy được mật khẩu thực tế của người dùng ở dạng văn bản rõ từ cookie, ngay cả khi nó đã được băm. Các phiên bản băm của danh sách mật khẩu nổi tiếng có sẵn trực tuyến, vì vậy nếu mật khẩu của người dùng xuất hiện trong một trong những danh sách này, việc giải mã băm đôi khi có thể đơn giản như chỉ cần dán băm vào công cụ tìm kiếm. Điều này chứng minh tầm quan trọng của muối trong mã hóa hiệu quả.

\* ĐẶT LẠI MẬT KHẨU NGƯỜI DÙNG

Trên thực tế, một số người dùng sẽ quên mật khẩu của họ, vì vậy, việc có một cách để họ đặt lại mật khẩu là điều thường thấy. Vì xác thực dựa trên mật khẩu thông thường rõ ràng là không thể trong trường hợp này, các trang web phải dựa vào các phương pháp thay thế để đảm bảo rằng người dùng thực sự đang đặt lại mật khẩu của chính họ. Vì lý do này, chức năng đặt lại mật khẩu vốn nguy hiểm và cần được triển khai một cách an toàn.

Có một số cách khác nhau để triển khai tính năng này, với các mức độ dễ bị tấn công khác nhau.

- GỬI LẠI MẬT KHẨU QUA EMAIL

Không cần phải nói rằng việc gửi cho người dùng mật khẩu hiện tại của họ không bao giờ có thể thực hiện được nếu một trang web xử lý mật khẩu một cách an toàn ngay từ đầu. Thay vào đó, một số trang web tạo mật khẩu mới và gửi mật khẩu này cho người dùng qua email.

Nói chung, việc gửi mật khẩu cố định qua các kênh không an toàn là điều nên tránh. Trong trường hợp này, tính bảo mật phụ thuộc vào việc mật khẩu đã tạo hết hạn sau một thời gian rất ngắn hoặc người dùng thay đổi lại mật khẩu ngay lập tức. Nếu không, cách tiếp cận này rất dễ bị tấn công theo kiểu trung gian.

Email cũng thường không được coi là an toàn vì hộp thư đến vừa cố định vừa không thực sự được thiết kế để lưu trữ thông tin bí mật một cách an toàn. Nhiều người dùng cũng tự động đồng bộ hộp thư đến của họ giữa nhiều thiết bị trên các kênh không an toàn.

- ĐẶT LẠI MẬT KHẨU BẰNG URL:

Một phương pháp mạnh mẽ hơn để đặt lại mật khẩu là gửi một URL duy nhất cho người dùng để đưa họ đến trang đặt lại mật khẩu. Các triển khai kém an toàn hơn của phương pháp này sử dụng một URL có tham số dễ đoán để xác định tài khoản nào đang được đặt lại, ví dụ:

<http://vulnerable-website.com/reset-password?user=victim-user>

Trong ví dụ này, kẻ tấn công có thể thay đổi tham số người dùng để tham chiếu đến bất kỳ tên người dùng nào mà chúng đã xác định. Sau đó, chúng sẽ được đưa thẳng đến một trang nơi chúng có khả năng đặt mật khẩu mới cho người dùng tùy ý này.

Một cách triển khai tốt hơn cho quy trình này là tạo một mã thông báo có entropy cao, khó đoán và tạo URL đặt lại dựa trên mã thông báo đó. Trong trường hợp tốt nhất, URL này sẽ không cung cấp bất kỳ gợi ý nào về mật khẩu của người dùng nào đang được đặt lại.

<http://vulnerable-website.com/reset-password?token=a0ba0d1cb3b63d13822572fcff1a241895d893f659164d4cc550b421ebdd48a8>

Khi người dùng truy cập URL này, hệ thống sẽ kiểm tra xem mã thông báo này có tồn tại ở phần cuối hay không và nếu có thì mã thông báo này sẽ đặt lại mật khẩu của người dùng nào. Mã thông báo này sẽ hết hạn sau một thời gian ngắn và bị hủy ngay sau khi mật khẩu được đặt lại.

Tuy nhiên, một số trang web không xác thực lại mã thông báo khi biểu mẫu đặt lại được gửi. Trong trường hợp này, kẻ tấn công có thể chỉ cần truy cập biểu mẫu đặt lại từ tài khoản của chính họ, xóa mã thông báo và tận dụng trang này để đặt lại mật khẩu của người dùng tùy ý.

Nếu URL trong email đặt lại được tạo động, điều này cũng có thể dễ bị đầu độc đặt lại mật khẩu. Trong trường hợp này, kẻ tấn công có khả năng đánh cắp mã thông báo của người dùng khác và sử dụng nó để thay đổi mật khẩu của họ.

\* THAY ĐỔI MẬT KHẨU NGƯỜI DÙNG

Thông thường, việc thay đổi mật khẩu của bạn bao gồm nhập mật khẩu hiện tại và sau đó nhập mật khẩu mới hai lần. Các trang này về cơ bản dựa trên cùng một quy trình để kiểm tra xem tên người dùng và mật khẩu hiện tại có khớp với nhau như trang đăng nhập thông thường không. Do đó, các trang này có thể dễ bị tấn công bằng các kỹ thuật tương tự.

Chức năng thay đổi mật khẩu có thể đặc biệt nguy hiểm nếu nó cho phép kẻ tấn công truy cập trực tiếp mà không cần đăng nhập với tư cách là người dùng nạn nhân. Ví dụ: nếu tên người dùng được cung cấp trong trường ẩn, kẻ tấn công có thể chỉnh sửa giá trị này trong yêu cầu để nhắm mục tiêu vào người dùng tùy ý. Điều này có khả năng bị khai thác để liệt kê tên người dùng và mật khẩu tấn công bằng cách dùng vũ lực.